# Supporting Applications in a Mobile Multihop Radio Environment Using Route Diversity - Part I: Non-real Time Data*

**Nitin Gogate** **Shivendra S. Panwar**

Department of Electrical Engineering
Polytechnic University
6 Metrotech Center, Brooklyn, NY 11201

## Abstract

This paper motivates the need for the Multiple Path Transport (MPT) of information in a multihop mobile radio network for supporting non-real time applications. Typical applications will need a higher bandwidth/higher reliability connection than that provided by current mobile networks. For supporting these applications a mobile node may need to set up and use multiple paths to the desired destination, either simply because of the lack of raw bandwidth on a single channel or because of its poor error characteristics, which reduces its effective throughput. In the context of this work, the principal reasons for considering such an architecture are providing high bandwidth and a more robust end-to-end connection. We describe a protocol architecture that addresses this need and, with the help of simple simulation models, we show that the delay and throughput performance of multiple path schemes is significantly better than that of a conventional scheme in which one session makes use of one path. For the data applications considered in this work, there is an additional advantage of security because tapping any one path does not give access to the complete information.

## 1 Previous Work

The problem of two communicating entities using multiple paths has been considered widely in the literature in various contexts. The earliest reference to multiple path[1] transport, referred to as *dispersity routing*, is in Dr. Maxemchuck's Ph.D. dissertation [1]. For example, a channel coding scheme using multiple parallel paths was considered in [2], which improved the fault tolerance of digital communication networks, while multiple parallel connections on different paths were set up to increase the maximum throughput between a pair of nodes in [6].

An important issue associated with communications using multiple paths is that of resequencing. As the traffic between a typical pair of end nodes follows different paths, which have different speeds (available bandwidth) and have different number of hops (entailing varying amounts of propagation delay and fixed processing delay), packets belonging to a session may arrive out of order at the destination node. The packets arriving out of order may have to wait in a special buffer called the resequencing buffer, before they can be delivered in order to the destination process. Some additional amount of delay is incurred due to this wait in the resequencing buffer. Most of the models considered in [11, 12, 18, 19] are of the source node, that is at the edge of a network, or of a single hop.

An exhaustive survey and comparison of various schemes to improve TCP performance over a wireless link can be found in [17]. These methods concentrate on mainly an internetwork scenario with only the last hop (link) being the wireless hop. In this work our emphasis is on the end-to-end throughput and delay performance and we can use any suitable method in [17] that gives us a reasonable link level performance. Thus, our work augments previous studies on TCP over a wireless link using a single path.

Finding multiple paths between a source-destination pair is not the focus area of this work. The need for source routing schemes in ad-hoc packet radio networks has been motivated in [20]. The source routing schemes could be used to find more than one route to the desired destination and the discovered routes could then be used for the MPT scheme. In this work, we concentrate mainly on traffic allocation and delay-throughput performance.

In the context of this work, apart from the obvious advantage of having a higher bandwidth connection, the other advantages of the MPT scheme are: a reduction in the traffic burstiness seen by individual paths, better adaptability to varying radio channel quality, and an inherent robustness to channel errors and link failures. It

*This work was supported in part by the New York State Center for Advanced Technology in Telecommunications, Polytechnic University, Brooklyn, New York.

[1] In this work, we use path and route interchangeably. Hence, in our definition of path/route diversity, two communicating entities make use of more than one path/route to send information to each other.

is intuitive to propose a protocol architecture which supports a generic multiple path communication and allows a multimedia application to utilize different physical or virtual paths. Our work differs from the previous work on MPT in that we consider it in a wireless network scenario where it is even more pertinent because of the limited bandwidth and the error characteristics of a single channel. Also, we address the MPT in protocol context, and consider some transport layer protocol issues.

# 2 Network and Protocol Model

The network scenario we assume is that of a CDMA network, with each mobile equipped with multiple transceivers, operating in a multihop packet radio environment. Although, we assume a multihop radio network scenario, in conventional cellular networks this corresponds to a mobile node capable of talking to either more than one base station or the same base station using multiple codes. Typically, each mobile can use receiver-directed or link-directed codes [10]. Thus the maximum bandwidth available to the application is the basic channel rate times the number of transceivers (codes).

In general, traffic splitting could be done at any layer of the protocol stack, for example physical, data link, and network (e.g. Inverse Multiplexing is being standardized by ATM-PHY group in the ATM Forum). We will consider traffic splitting at the network layer and above. Also, given the widespread use of the TCP/IP protocol stack, it is imperative that a new scheme should work well with existing transport and network level protocols. The layered protocol model is shown in Figure 1. Option (A) in the Figure 1 corresponds to adding a new layer (called Meta-TCP) with the functions of distributing the traffic from one application (source) to multiple TCP connections, which then follow different paths and of reordering the packets from these different connections back into single stream at the destination. The Meta-TCP layer could be thought of as a session layer entity which provides a uniform interface between the application layer and the underlying transport layer or could also be embedded in the application itself. Option (A) is more suitable for a transport layer which relies on algorithms that treat reception of multiple out of order packets as an indication of bad state. TCP is an example of this kind of protocol, since it estimates the route quality by doing round trip time measurements and has special treatment for out of order packets. In option (A) as we are running separate windows on each path, traffic spreading does not interact with the TCP algorithms. Although TCP can handle the out-of-order packets, it may not be amenable to the additional out-of-order packets introduced by the traffic splitting mechanism at the sender. With TCP as a transport layer, it does not seem feasible to run a single window on all the paths transparently, for example as in option (B) where the traffic is distributed at the network layer. This is because of the interaction of the out-of-order packets with the TCP algorithms (e.g. Round Trip Timing algorithm and Fast Retransmit algorithm). Such an interaction with Fast Retransmit algorithm was also observed in [17], where the link layer protocols did not attempt in-order delivery across the link. An option of performing the resequencing at the IP layer and then deliver the sequenced stream to the TCP is not feasible as there are no provisions to do this even in IPv6.

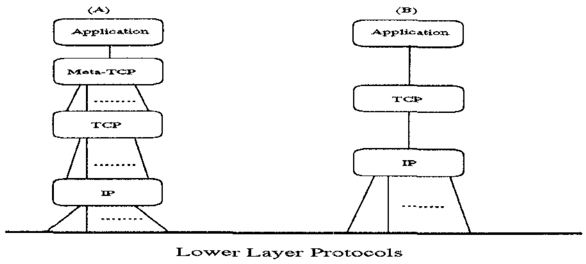


Figure 1: Layered Protocol Model.

The applications which we will consider are large file/image transfers or sending low/high resolution video streams in an environment with varying channel quality in terms of bit error rate and channel availability. In this work, we will deal with only the file transfer application. In a forthcoming paper, we will use the MPT scheme in conjunction with a *Multiple Description Coding* (MDC) scheme [16] for providing a robust image/video transfer. In the multiple description coding scheme, on the sender side, a coder breaks the signal into multiple substreams, each substream is then coded, packetized and sent on different path through the multihop radio network. Such a coding scheme is non-hierarchical in the sense that each subsignal contains sufficient information for the decoder to reconstruct an acceptable quality signal. At the receiver side the decoder produces acceptable quality signal from any single substream and each additionally received substream contributes to enhanced image/video quality.

## 2.1 Transport of data

For file transfer, which needs reliable transport of data, we will consider TCP as an example transport protocol. With the emergence of the world wide web (WWW) over the past few years, network traffic characteristics have changed dramatically. The amount of data transferred per request and the traffic burstiness has also increased. In the context of WWW/HTTP, where a set of files is transferred per request (when a hypertext link is followed), we can use the multiple route transport scheme even at a layer above the TCP layer. That is, in this scenario the traffic could be split even on a file by file basis. This is another level of granularity. We define the "Granularity" of MPT as the smallest unit of information allocated to each route. For example, theoretically, granularity could be a bit, a byte, a link layer packet, an IP packet, a TCP segment or a session. Of course, the coarser the granularity the better it is from the resequencing view point, but the disadvantage with coarse granularity is that we lose in the increased burstiness and queueing delay seen by each path.

# 3 Simulation Model

This section describes the simulation models which we use in analyzing the performance of TCP and Meta-TCP over multiple paths. Also, various parameters related to the traffic, model and link are discussed.

One can set up a simple model of two application level processes communicating on two separate paths in one case and in the other case sharing the two paths. This is a higher layer version of the problem considered in [6]. For this problem, we expect that the reduced burstiness of the source (as seen by each path), may give us better performance in terms of end-to-end delay. The system schematic is shown in Figure 2. For this set-up, an application layer entity opens two (in general, the number of sockets opened could be related to the number of available paths which can be used) separate socket (logical TCP connections) connections with the destination application layer entity. The traffic is then spread over these two logical connections. In general, the paths could consist of several wireless hops. In the simulation, both the paths are identical in terms of available bandwidth and the number of hops. Each path consists of three wireless hops. In the simulations, in order to study the effect of the wireless channel, we have set the parameters such that the losses are mainly due to channel errors.

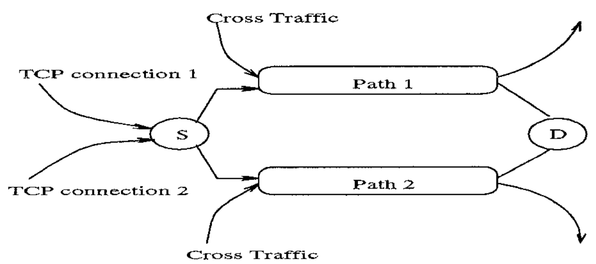


Figure 2: A simple end-to-end model.

When packets belonging to one application stream are distributed to multiple TCP socket connections, packets arriving out of order at the destination Meta-TCP layer have to wait in a special buffer called the resequencing buffer. This wait contributes additionally to the end-to-end delay. The Meta-TCP layer, traffic distribution and the resequencing buffer are implemented in the application layer in the simulation model.

## 3.1 Channel Model

There is an extensive literature on wireless channel modeling based on theory as well as measurements, both in the indoor and outdoor (urban, suburban and rural) environments [13, 14]. As our primary focus is to study MPT schemes, we chose to qualitatively model the bursty error nature of the wireless channel as described below. We model the radio link by a two state Markov model with the two states corresponding to the link being in either a good state or a bad state. Further, we assume the presence of some error correcting code at the Data link layer, so the channel can be qualitatively modeled by the two state Markov model. In the good state we assume that a packet is lost with some low probability $p$ (or the BER is so low that most errors are corrected at the link layer), whereas in the bad state the packet is lost with probability $q$. We use a value of 0.005 for $p$ and 1 for $q$. This is in accordance with the Jakes model [13] which is a commonly used model. Based on the Doppler frequency and how deep a fade (below the average level) the coding can tolerate, one can calculate the average fade duration and the level crossing rate. From these we can then calculate the transition rates between the good and the bad states. We consider a radio channel operating at 2.0 Mbps. This is the raw bandwidth of any single channel. Typically, the bandwidth seen at the application layer is much less than 2.0 Mbps due to packet losses at the the link layer and retransmissions at the transport layer.

# 4 Results and Discussion

In the simulation results presented below, we consider a very simple end-to-end model with two paths. The paths are being used transparently (i.e. no information is kept/used regarding the "quality" of the path) either on a per session basis or on a round-robin basis. Note that the round-robin scheme used here is slightly different that the conventional round-robin in that it sends $N$ consecutive packets on a path rather than one. For this study we have used a value of $N$=5. This is done in order to reduce the out-of-order packets and the resequencing delay at the receiver. The main aim of this study is to investigate the burst reduction properties of the multipath scheme, to estimate the resequencing delay at the receiver and to perform delay/throughput comparisons of the two schemes. We are in the process of running more simulations to study the effect of interfering traffic on the end-to-end delay and throughput.

First we consider a large file transfer application. We assume that the information flow is bidirectional, that is data is being transferred in both directions once the connection is established. Further, the MSS (Maximum Segment Size) of TCP is set to 536 bytes, and the minimum and the maximum limits on RTO are set to 200 ms and 800 ms respectively. The delayed-ACK interval is set to 50 ms. Further, the receiver buffer size is set to 32 Kbytes. We compare the performance of the round-robin and the session routing schemes, for various error rates. The packet error rate is controlled by the average duration of the good and the bad states. In the file transfer simulations each sender initiates a large file transfer (of size 1Mbyte). As there are two application processes at the source and destination the total bulk of data transferred over both the paths is 4 Mbytes. For the throughput studies we assume a bidirectional flow of data from a greedy source which always has a packet to send when allowed by the TCP windowing mechanism. The file transfer and the throughput results are presented in Tables 1 and 2 respectively. We can see that the round-robin schemes performs better than the session based routing scheme. The numbers in parentheses represent the 95% confidence interval.

As a starting point, we approximate the bulk interactive model as follows. There are requests whose inter-arrival time is exponential and each request transmits a batch of $i$ packets, where $i$ is a uniformly distributed random variable between 1 and some maximum batch-size. The inter-request time could be thought of as time taken by the user to absorb the previous information (in case of WWW) and request new information. Although this traffic model is rather simple compared to the empirical study reported in [21], we feel that it may give us some insight. The results, for different average batch sizes, are presented in Table 3. It can be seen that as the average batch size is reduced (i.e., traffic becomes smoother) the resequencing delay starts dominating and the round-robin performance becomes worse.

Another important note should be made regarding the fact that we have not used any means of improving TCP performance over the wireless hop as described in [17]. We use a native mode TCP (which is similar to the TCP-Tahoe version) connection operating on an end-to-end basis. The improvement in the performance of MPT scheme may be due to quicker opening of multiple windows (in the slow-start phase) and due to the reduction in burstiness. It is clear that implementing these "intelligent" schemes at each radio hop would significantly improve the overall end-to-end performance. We have avoided doing that so far because we want to check the end-to-end performance of the native mode TCP for the two schemes as a benchmark. We also propose to evaluate the MPT scheme performance for doing adaptive routing based on some measure of "path quality", such as packets in error or TCP window size or other information available at the source.

The results presented in this study are for a TCP implementation with the above-mentioned parameters and without the fast retransmit and Karn's Algorithm [15]. We expect that implementing these algorithms will further improve the the performance for both the cases considered here.

### 4.1 Performance with a reliable link layer protocol

In this section we analyze the performance of TCP with a reliable link layer protocol. The link layer protocol handles the retransmissions locally and we show that with proper timer settings the link layer retransmissions do not interact with the TCP layer timers. Further, as the packet transmission time is large as compared to the typical propagation delays we are considering, we can set the link level window size to a low value. For the set of parameters we have tried and for the setup we considered, initial simulation results show a significant improvement in the end-to-end performance. These results are presented in Table 4.

In general, the interaction between the TCP layer timers and the local retransmissions is avoided mainly because of the coarse timers of most of the TCP implementations. In simulations, though the timers are not coarse, still the above interaction is avoided because of the high value of $RTO_{min}$ as compared to the Link Layer timers. Also, as the link level window size is small and because of the selective retransmissions at the link layer, there are less out-of-order packets. Hence, again algorithms like fast retransmit are not adversely affected.

## 5 Conclusions

We have motivated the need for multiple route communication in multiple hop radio networks for transport of non-real time data. We present a protocol model which handles this paradigm. With the help of simple simulation models we compare the delay and throughput performance of multiple route schemes to that of a conventional scenario in which all the traffic follows a single route. In general, the MPT protocol architecture provides a generic framework for transporting a variety of information. As an ongoing work, we also propose to use the MPT scheme with Multiple Description Coding (MDC) to transport real-time video streams to a destination in a packet radio network.

## References

[1] N.F. Maxemchuk, "Dispersity Routing in Store and Forward Networks," Ph.D. Dissertation, University of Pennsylvania, May 1975.

[2] E. Ayanoglu, I. Chih-Lin, R.D. Gitlin and J.E. Mazo, "Diversity Coding: Using Error Control for Self-healing in Communication Networks," *Proc. of IEEE INFOCOM'90*, pp. 95-104, June 1990.

[3] V. Jacobson, "Congestion Avoidance and Control," *SIGCOMM 88*, August 1988.

[4] J. Postel, "Transmission Control Protocol - DARPA Internet Program Protocol Specification," *RFC 793*, September 1981.

[5] V. Jacobson, R. Braden and D. Borman, "TCP extensions for high performance," *RFC 1323*, May 1992.

[6] R. Krishnan and J.A. Silvester, "Choice of Allocation Granularity in Multipath Source Routing Schemes," *Proc. of IEEE INFOCOM,* pp. 322-329, March 1993.

[7] S. Nanda, R. Ejzak and B. Doshi, "A Retransmission Scheme for Circuit-Mode Data on Wireless Links," *IEEE JSAC*, Vol. 12, No. 8, pp. 1338-1352, Oct. 1994.

[8] A. DiSimone, M.C. Chuah and O.C. Yue, "Throughput performance of Transport Layer Protocols over Wireless LANs, *IEEE Globecom*, December 1993.

[9] N. Shacham and P. King, "Architectures and Performance of Multichannel Multihop Packet Radio Networks," *IEEE JSAC*, vol. 5, no. 6, July 1987.

[10] L. Hu, "Distributed Code Assignments for CDMA Packet Radio Networks," *IEEE/ACM Tran. on Networking*, Vol. 1, no. 6, pp. 668-677, Dec. 1993.

[11] S. Chowdhury, "Distribution of the Total Delay of Packets in Virtual Circuits," *IEEE INFOCOM*, pp. 911-918, April 1991.

[12] A. Jean-Marie and L. Gun, "Parallel Queues with Resequencing," *Journal of ACM*, Vol. 40, no. 5, pp. 1188-1208, 1993.

[13] W. C. Jakes, "Microwave Mobile Communications," reissued by IEEE press, 1993.

[14] G. L. Stuber, "Principles of Mobile Communication," Kluwer Academic Publishers, 1996.

[15] W. Richard Stevens, "TCP/IP Illustrated, Volume 1," Addison Wesley, Inc.

[16] Y. Wang and D. Chung, "Non-Hierarchical Signal Decomposition and Recovery for Image Transmission in Wireless Networks," in *Proc. of MoMuC3*. Sept. 1996. Princeton, NJ.

[17] H. Balakrishnan, V. Padmanabhan, S. Seshan and R. Katz, "A Comparison of Mechanisms for Improving TCP performance over Wireless Links," *Proc. of ACM SIGCOMM*, Stanford, 1996.

[18] N. Gogate and S. S. Panwar, "On a Resequencing Model for High Speed Networks," In the Proc. of IEEE INFOCOM, June 1994.

[19] N. Gogate and S.S. Panwar, "Assigning Customers to Two Parallel Servers with Resequencing," *Proc. of the CISS,* Johns Hopkins University, 1995.

[20] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing,* edited by T. Imielinski and H. Korth, pp. 153-181, Kluwer Academic Publishers, 1996.

[21] Bruce A. Mah, "An Empirical Model of HTTP Network Traffic," *Proc. of IEEE INFOCOM*, 1997.

| $\overline{G}$ (ms) | $\overline{B}$ (ms) | Pkt loss Prob/hop | Time (sec) Round-robin | Time (sec) Session-based |
|---|---|---|---|---|
| 1000 | 10 | 1.71 % | 20.02 [17.45,22.59] | 32.29 [27.68,36.89] |
| 100 | 1 | 3.73 % | 91.24 [82.89,99.59] | 170.52 [162.37,178.68] |
| 100 | 10 | 11.61 % | 204.91 [198.61,211.21] | 390.59 [364.67,416.51] |

Table 1: Time required (in seconds) to transfer a large file (Size = 1 Mbytes) vs. Packet loss probability.

| $\overline{G}$ (ms) | $\overline{B}$ (ms) | Pkt loss Prob/hop | Throughput Round-robin | Throughput Session |
|---|---|---|---|---|
| 1000 | 10 | 1.71 % | 121.39 [112.07,130.70] | 65.09 [63.91,66.28] |
| 100 | 1 | 3.73 % | 22.68 [22.21,23.14] | 11.50 [11.25,11.76] |

Table 2: Throughput (packets/sec) vs. Packet loss prob.

| $\overline{S}$ | $1/\lambda$ (sec) | Avg. delay Round-robin (sec) | Avg. delay Session(sec) |
|---|---|---|---|
| 50 | 10 | 0.929 [0.810,1.049] | 1.292 [1.190,1.395] |
| 25 | 5 | 0.507 [0.482,0.532] | 0.661 [0.589,0.732] |
| 5 | 1 | 0.162 [0.149,0.175] | 0.180 [0.154,0.205] |
| 1 | 0.2 | 0.055 [0.0521,0.0588] | 0.048 [0.0426,0.0540] |

Table 3: Interactive performance (Average delay) for various batch sizes ($\overline{S}$) with channel average good ($\overline{G}$) and bad ($\overline{B}$) durations 1000 and 10 ms respectively.

| $\overline{S}$ | $1/\lambda$ (sec) | Avg. delay Round-robin (sec) | Avg. delay Session (sec) |
|---|---|---|---|
| 50 | 10 | 0.397 [0.385,0.409] | 0.563 [0.549,0.576] |
| 25 | 5 | 0.272 [0.266,0.278] | 0.383 [0.373,0.394] |
| 5 | 1 | 0.1156 [0.1142,0.1170] | 0.1370 [0.1348,0.1392] |
| 1 | 0.2 | 0.03665 [0.03627,0.03702] | 0.03595 [0.03560,0.03630] |

Table 4: Preliminary results for Interactive performance with a reliable link layer protocol, for various batch sizes ($\overline{S}$), with $\overline{G} = 1000$ and $\overline{B} = 10$ ms respectively.